Software Requirements Specification

for

Facial Motion-tracking Based Mouse Pointer Controller for Disabled

Version 1.0 approved

Prepared by

Umar Tariq Wani	102083065
Divyanshu Shukla	101903487
Aarohan Verma	101903497
Prachi Mangewala	101903493
Rahul Vats	101953018

BE Third Year (COE19), TIET

Submitted to: - Dr. Arwinder Kaur (Lab Instructor, Software Engineering, TIET)

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
First Draft	3rd October 2021	First Draft	1.0

1. Introduction

1.1 Purpose

This project analyzes the biometric identification and tracking related technologies of human-computer interaction. Based on face detection algorithm, we propose a position-based head motion detection algorithm, which does not depend on the specific biometric identification and tracking. It uses feature classification method to detect eye opening and closing actions. We also design a software system to operate computer by image detection of head and eye movements. The combinations of head and eye movements, are mapped to various mouse events, including move, click and drag, and so on.

1.2 Document Conventions

"Project" refers to *"Facial Motion-tracking Based Mouse Pointer Controller for Disabled"*

1.3 Intended Audience and Reading Suggestions

This document is intended for the developers and beta users.

1.4 Project Scope

This system can be used for the upper limb disabled who failed to use the traditional mouse and keyboard. Furthermore, it can also be used for general computer users to do neck rehabilitation training, computer somatic games, etc.

1.5 References

[1] P. Viola and M. Jones, "Computer Vision and Pattern Recognition" 2001

[2] T. Hutchinson, K. P. White Jr., W. N. Martin, K. C. Reichert, and L. A.Frey, "Human-computer interaction using eye-gaze input" 1989.

IEEE PAPERS

[1] D. G. Evans, R. Drew, and P. Blenkhorn, "Controlling mouse pointer position using an infrared head-operated joystick,"2000.

[2] M. Betke, J. Gips, and P. Fleming, "The Camera Mouse: Visual Tracking of Body Features to Provide Computer Access for People With Severe Disabilities," On Neural Systems And Rehabilitation Engineering, March 2002.

[3] M. Nabati and A. Behrad, "Camera Mouse Implementation Using 3D Head Pose Estimation by Monocular Video Camera and 2D to 3D Point and Line Correspondences," 2010 5th International Symposium on Telecommunications (IST'2010), 2010.

[4] Y. L. Chen, F. T. Tang, W. H. Chang, M. K. Wong, Y. Y. Shih, and T.S. Kuo, "The new design of an infrared controlled human-computer interface for the disabled," Dec. 1999.

WEBSITES

[1] www.disabled-world.com

[2] www.ncbi.nlm.nih.gov

[3] www.spd.org.sg

.

2. Overall Description

2.1 Product Perspective

The product uses the feature classification method to map the mouse pointer on the screen to the movements of head and eye in frames through a camera. The system analyzes the relationship between different combinations of the detected head and eye open and closing action, and then maps them to mouse events on the computer system. Our aim is to use this application mainly for the upper limb disabled who are unable to use the traditional mouse.

2.2 Product Features

The face detection is implemented by using Adaboost algorithm in our system, in which each frame of video streaming captured by a camera is input signal. Then we propose an algorithm for detection of head movements by analyzing face locations. We defined five motions as the basis of head movements, namely, standard head, head left, head right, head up, and head down

2.3 User Classes and Characteristics

There is a single user class, which is managed by the end user interacting with the system interface. It can interact as an alternative input to the mouse pointer driver installed in the system.

2.4 Operating Environment

The systems can run on any Windows or Linux based GUI Operating Systems with a web-cam support.

2.5 Design and Implementation Constraints

Web-cam support is compulsory for any graphical operating systems on which it is intended to run.

2.6 User Documentation

A proper *README.txt* shall be delivered along with the final version of the product.

2.7 Assumptions and Dependencies

- The OS is graphical
- It is either windows or Linux based for running executable file.
- There is pre-installed webcam support in the system.

3. System Features

While the system is running, if standard head and eyes are detected, *'Start'* option can activate the system.
After that, user can operate mouse by the head and eye movements.

The working process and principles of the system are described as follows:

(1) If standard head is detected, indicating the head is still, the mouse is in the state of idle

(2) In the state of idle, if the movement of head left is detected, the cursor will move left horizontally. If the user keeps the action of head left, the cursor keeps moving left. When the user moves back to standard head, the cursor stops moving and goes back to the idle state. Mouse cursor moving right, up and down have the same principle as moving left.

(3) In the state of idle, if closed eye is detected, the selected option of clicking is activated. In the state of selection, closing of the eye performs the selected option of left click, right click, double- click or scroll commands. Usually, the movement of closing eye lasts about 2 seconds. Opening eyes again makes the cursor go back to idle.

(4) Motion and selection can be switched through the closing eye movement.

(5) If the threshold gets wrongly mapped then the Reinitialize button refreshes the threshold and sets a new threshold based after pressing the Start option again.

(6) Stop option stops the system

4. External Interface Requirements

4.1 User Interfaces

Considering the problems with the current system, understanding the user’s needs and expectations the requirements for the proposed system is collected and found to be complete. Also, the system does not require any additional features that may cause delay in the release of the project. This makes our project functionally measurable, realistic and complete

4.2 Hardware Interfaces

1. Web camera

2. Laptop or a computer.

4.3 Software Interfaces

To be used efficiently, all computer software needs certain hardware components or other software resources to be present on a computer. These pre-requisites are known as system requirements and are often used as a guideline as opposed to an absolute rule.

Supported Operating Systems:

- Windows 10
- Supports both 32 bit as well as 64 bit OS.

The software requires following applications installed on the server machine:

- JDK1.6 or above
- NetBeans IDE 7.0 or above.
- OpenCV.

4.4 Communications Interfaces

Not Applicable

5. Other Nonfunctional Requirements

5.1 Availability and Supportability Requirements

The user does not need to register to the system before using it. It can be used by any Human due to the feature classification method used in the system. It is easy to use and available to all. This system is supported by any Java application. It can be used with any web camera. The accuracy of the system though may depend on the specifications of the camera.

5.2 Safety and Security Requirements

The architecture is entirely standalone, so there is no need of any external connections like internet connections, etc., which reduces completely the risk of any kind of data breach or virus attack.

5.3 Software Quality Attributes

Quality Of Service (QOS) is a major issue in desktop applications. In the proposed system, there is constant transfer of data between the peripheral (here webcam) and the system. The simplicity of the algorithm enhances the response time, thus bettering the QOS of the system

6.1 Activity Diagram

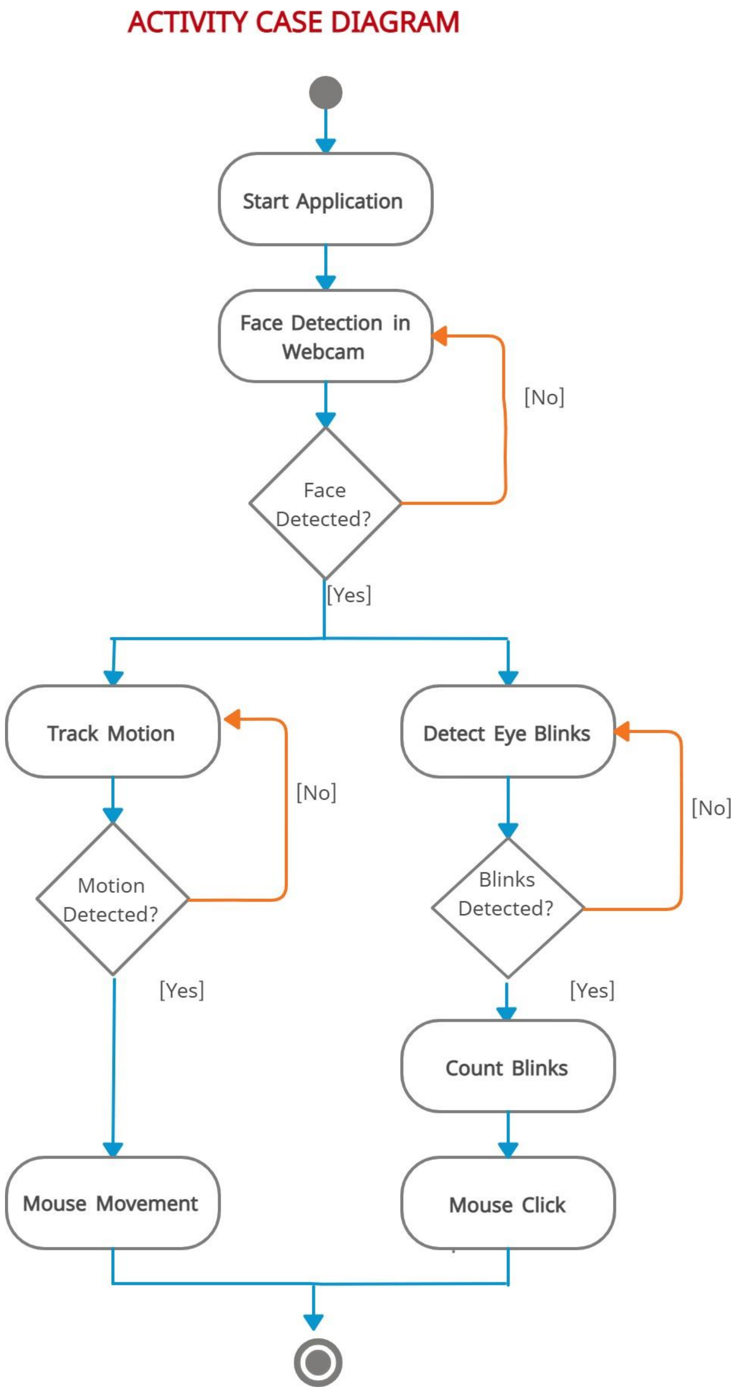


Figure 1

Description : Activity diagram at the expected final stage of development (Scope for change)

6.2 Use Case Diagram

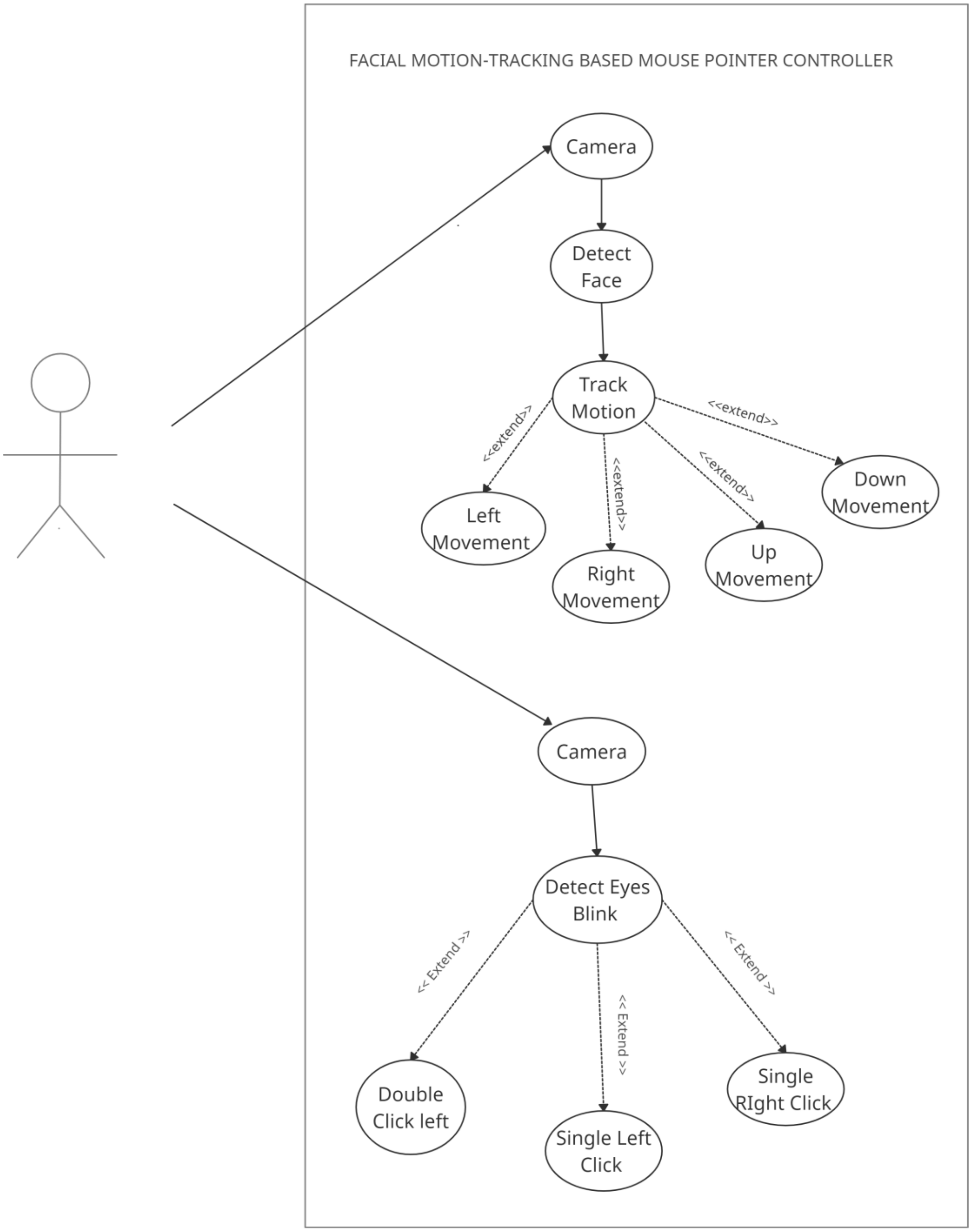


Figure 2

Description : Use case diagram at the expected final stage of development (Scope for change)

6.3 Entity Relationship Diagram

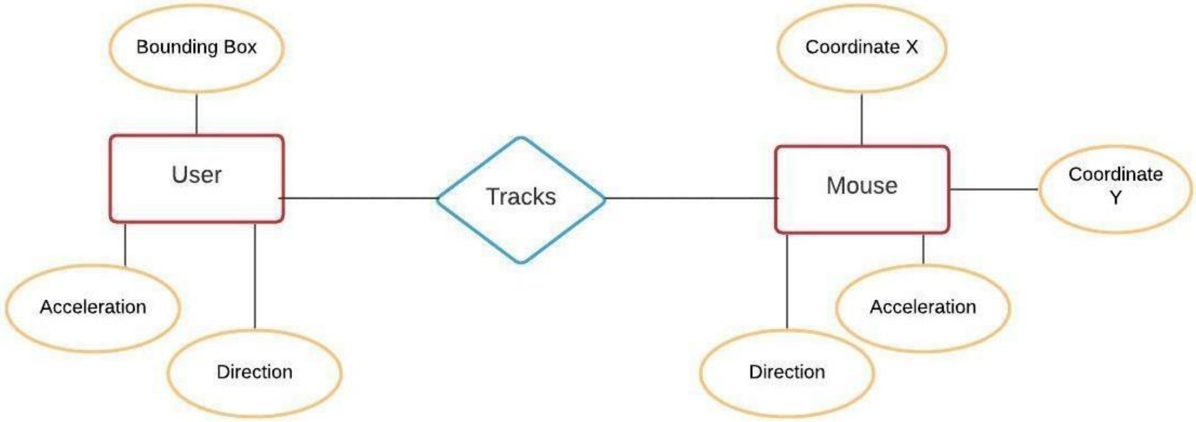


Figure 3

6.4 Sequence Diagram

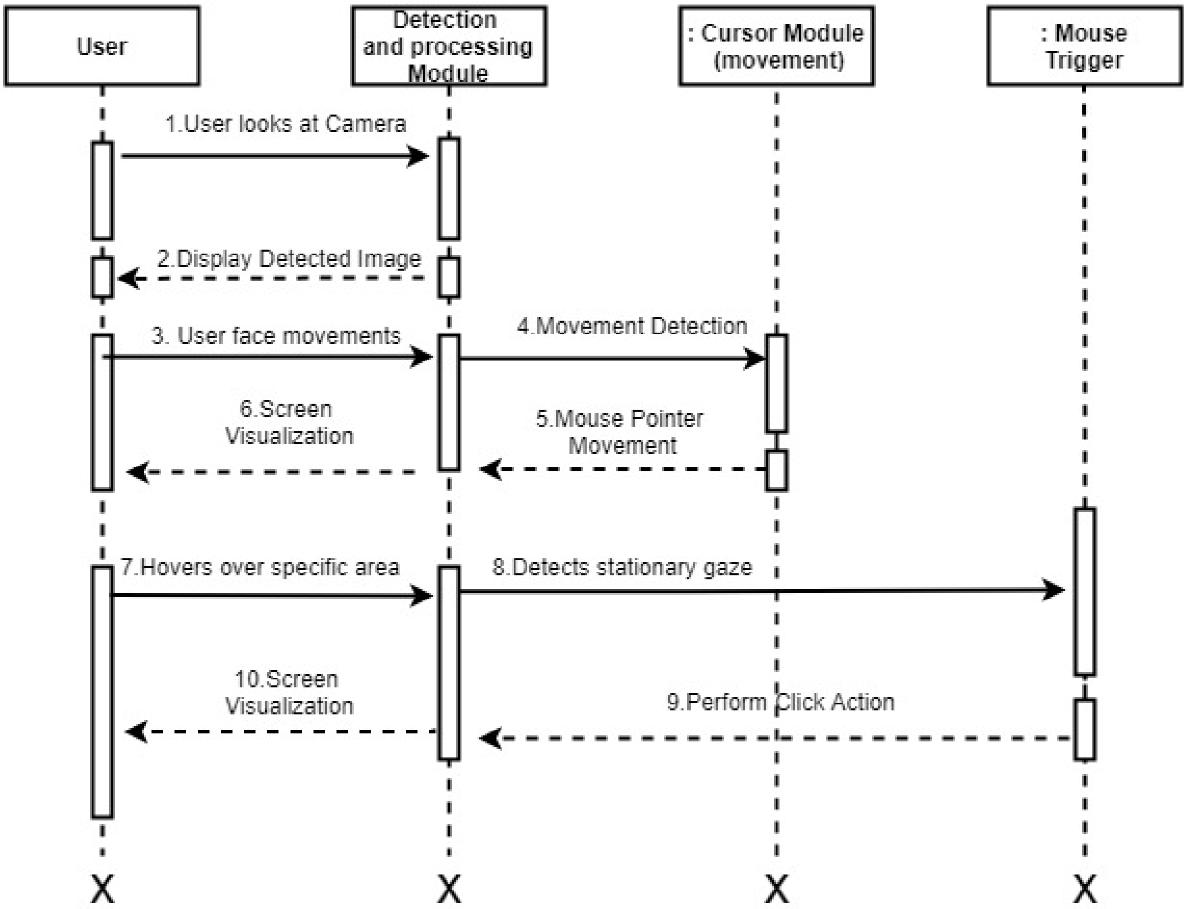


Figure 4

Description: Sequence diagram at current stage of development (Scope for additional features in future)

6.5 Collaboration Diagram

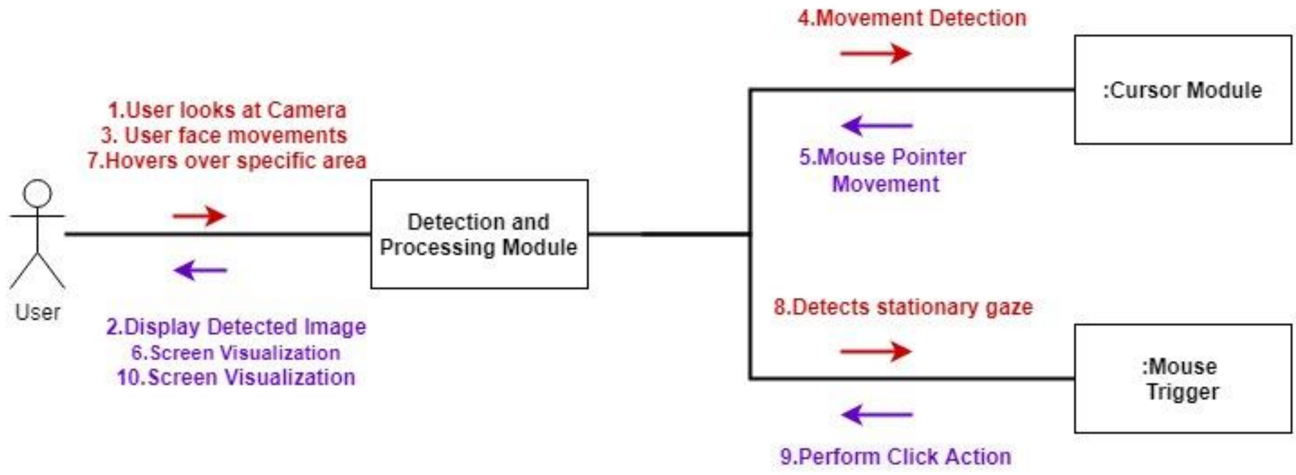
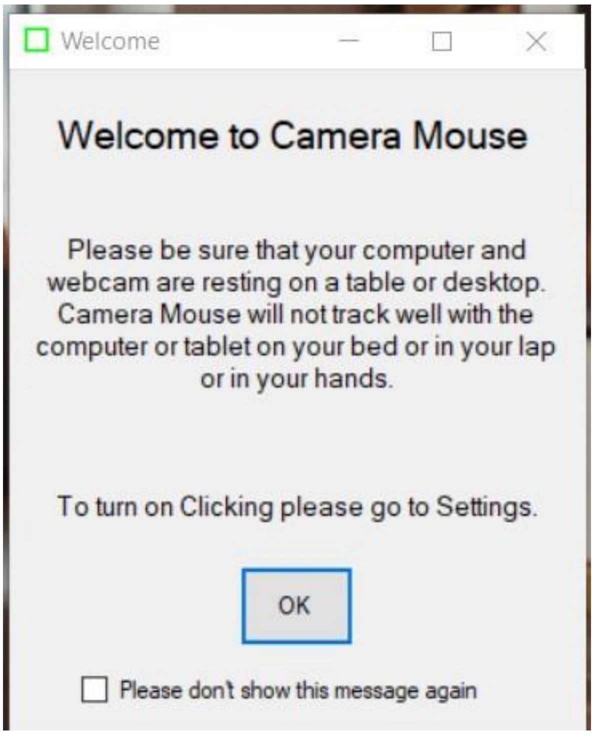


Figure 5

6.6 Graphical User Interface



Welcome to Camera Mouse

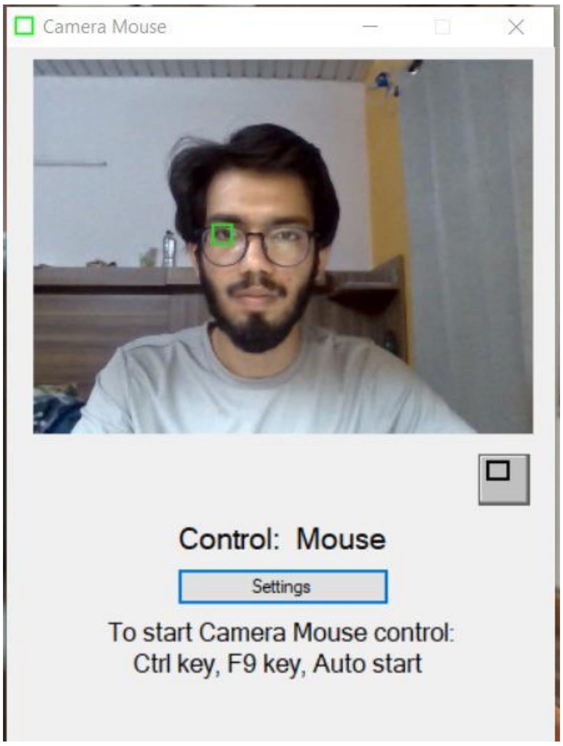
Please be sure that your computer and webcam are resting on a table or desktop. Camera Mouse will not track well with the computer or tablet on your bed or in your lap or in your hands.

To turn on Clicking please go to Settings.

☐ Please don't show this message again

Figure 6

Description: (Welcome screen and pre-initialisation message screen)



Control: Mouse

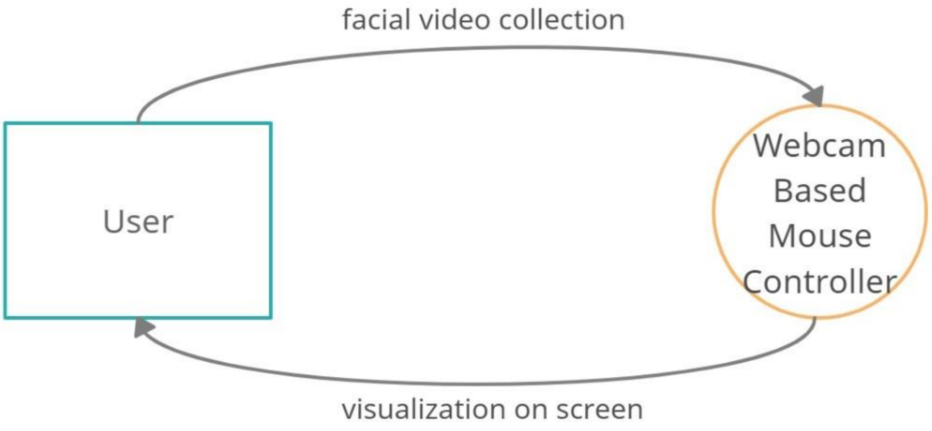
To start Camera Mouse control:
Ctrl key, F9 key, Auto start

Figure 7

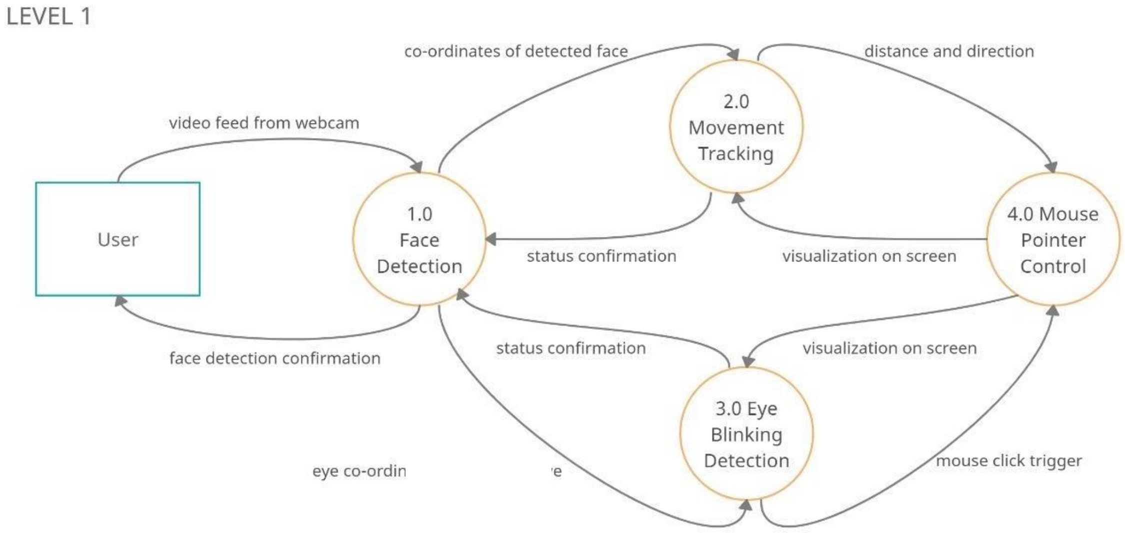
Description: Software GUI and settings option to calibrate the cursor with chosen organ (eye) movement, and adjust functioning of cursor and trigger ex. Speed, number of clicks, dwell time etc.

6.7 Data Flow Diagram

DFD Level 0

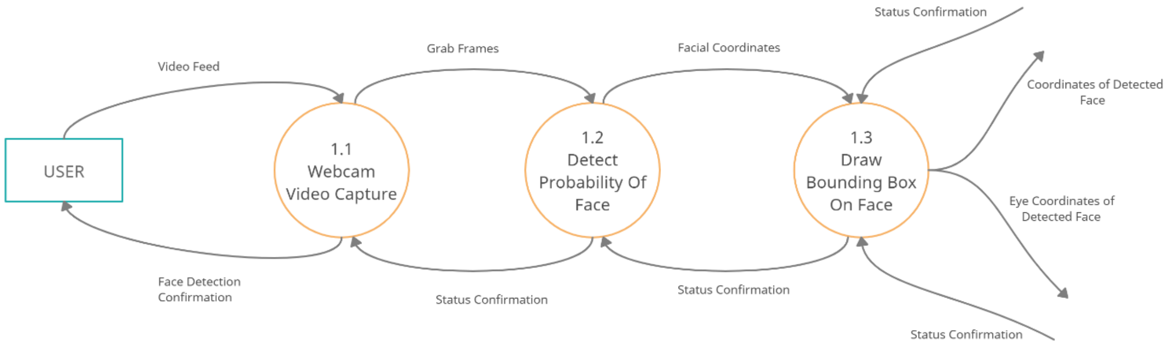


DFD Level 1

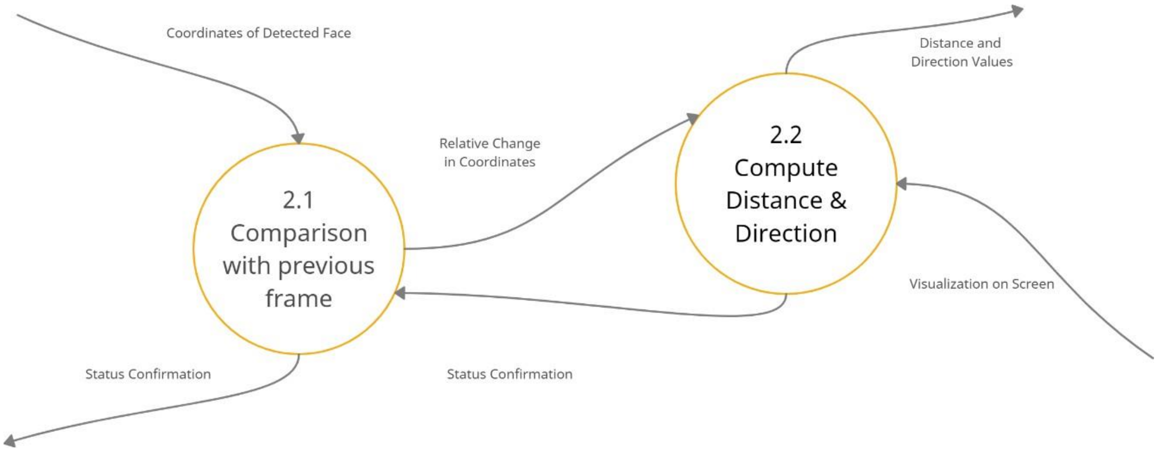


DFD Level 2

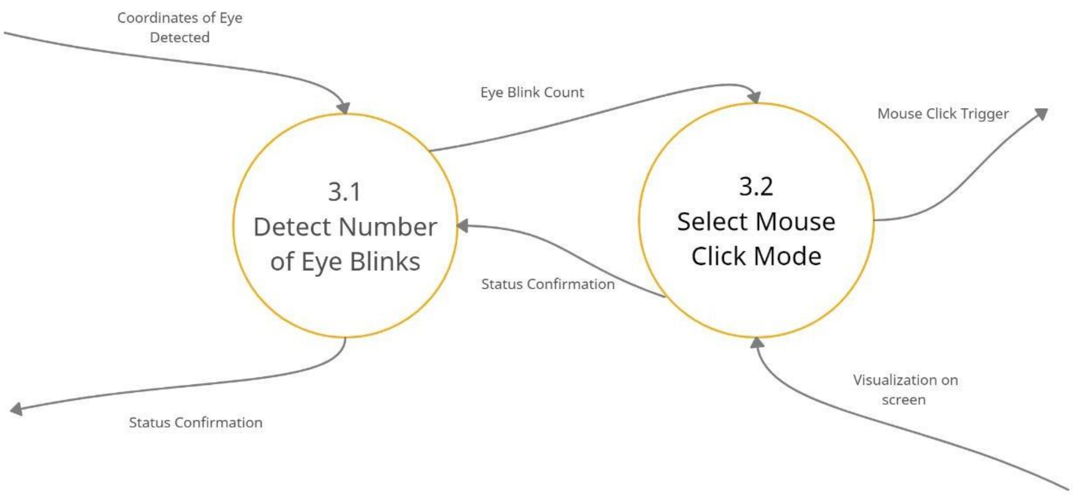
1.0


Status Confirmation
Grab Frames
Facial Coordinates
Video Feed
Coordinates of Detected Face
1.1 Webcam Video Capture
1.2 Detect Probability Of Face
1.3 Draw Bounding Box On Face
USER
Eye Coordinates of Detected Face
Face Detection Confirmation
Status Confirmation
Status Confirmation
Status Confirmation

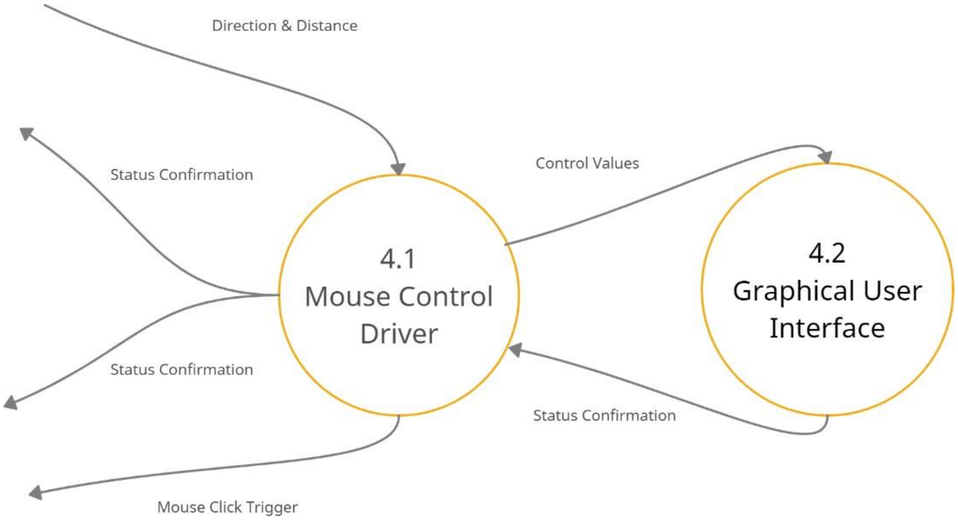
2.0


Coordinates of Detected Face
Distance and Direction Values
Relative Change in Coordinates
2.2 Compute Distance & Direction
2.1 Comparison with previous frame
Visualization on Screen
Status Confirmation
Status Confirmation

3.0


Coordinates of Eye Detected
Eye Blink Count
Mouse Click Trigger
3.1
Detect Number of Eye Blinks
3.2
Select Mouse Click Mode
Status Confirmation
Visualization on screen
Status Confirmation

4.0


Direction & Distance
Status Confirmation
Control Values
4.1
Mouse Control Driver
4.2
Graphical User Interface
Status Confirmation
Status Confirmation
Mouse Click Trigger

5

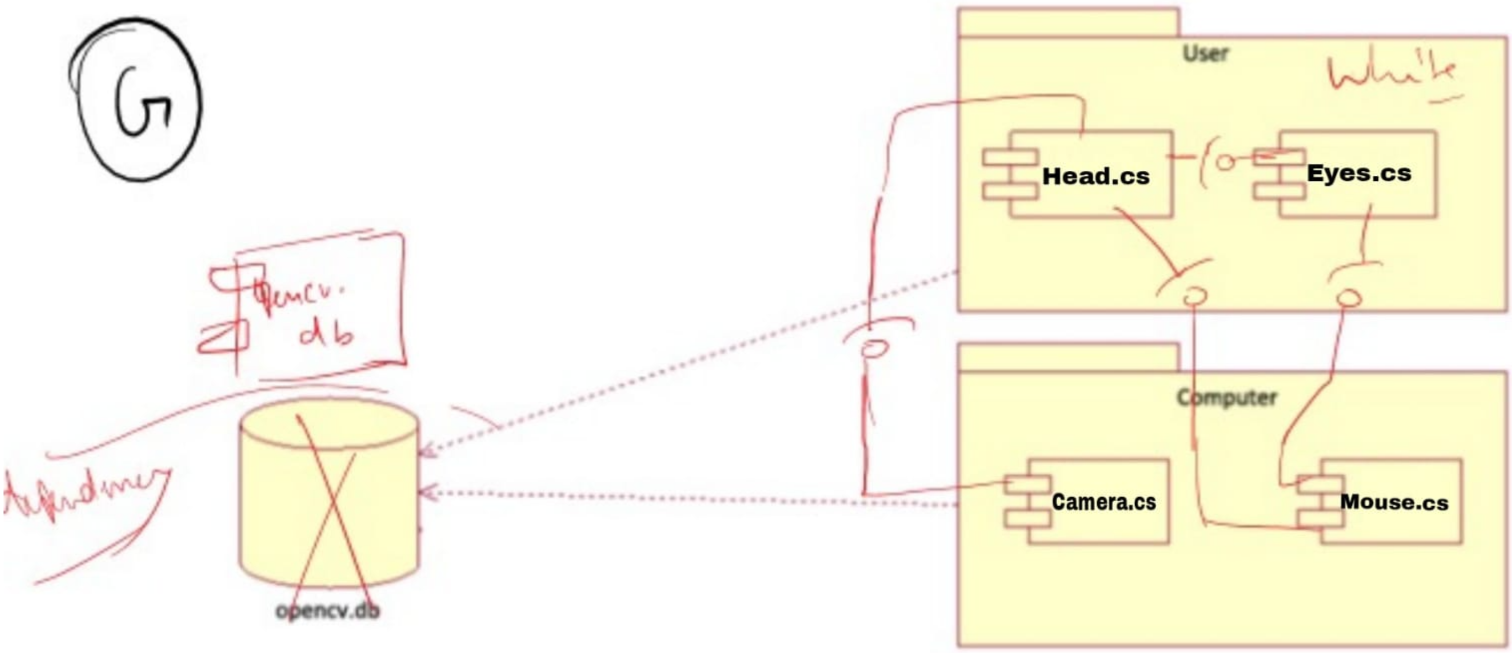


Fig. 4.11 Component diagram

Camera Mouse

Test Plan

Release 1.0

Author Details

Author(s)	
Name	**Title**
Divyanshu Shukla	Tester

Revision History

Version No.	Date (DD/MMM/YYYY)	Name	Description / Reason for Change(s)
1.0	3/Oct/2021	Divyanshu Shukla	New Document

Table of Contents

1 Objective

The objective of this document is to define the overall strategy and approach for Unit Testing and System Testing of Camera Mouse. This document describes how these two phases of testing will be organized, managed and executed.

2 Scope

The scope of this document is limited to describing unit test and system test activities for Camera Mouse

3 Definitions

Terms	Definition
ST	System Test
UT	Unit Test

4 Roles and Responsibilities

Roles and responsibilities, assigned to specific team members.

5 Test Phases

The following table defines the test phases to be conducted for testing and verification of Camera Mouse. The testing will ensure that the application functionality conforms to requirements described in the functional requirements specification.

Test Phase	Relevant Document(s)	Approach
Unit Test	• Software Requirements Specification Camera Mouse v 1.0	1. Unit Test Scripts will be created based on requirements described in Software Requirements Specification 2. Executed Test Scripts will be reviewed
System Test	• Software Requirements Specification Camera Mouse v 1.0	1. System Test Scripts will be created based on the requirements described in Software Requirements Specification 2. Executed Test Scripts will be reviewed

5.1 Unit Test

5.1.1 Prerequisites to Unit Test

1. Test Plan
2. SRS
3. Unit Test Scripts
4. Development environment with Camera Mouse Release

5.1.2 Overview, Scope and Objective

Unit Test will be carried out for Camera Mouse application to confirm that the code units conform to the unit level requirements defined in the FRS.

The unit test scripts will cover the following:

A. Manual testing will be performed by executing the unit test scripts in the development environment

   a. Tester instructions:

      i. Execute steps listed in the “Test Description” column in the unit test script. Compare outcome of each step with details documented in the “Expected Result” column.

      ii. After completion of Unit Test script execution, the tester is required to digitally sign along with date in the “Script Executed By” section

   b. Test Script Deviation and Deviation Resolution:

      i. For any test where the observed result does not match the expected result, will cause a deviation.

      ii. Any unresolved deviations will be documented with justification in the Validation Summary Report

   c. Testing Suspension

      i. Testing will be suspended between the identification of a deviation and its resolution.

   d. Reviewer instructions:

      i. Review the executed test scripts and ensure that the test script execution, documentation is correct and accurate.

B. The following code units will be tested during the unit testing of Camera Mouse:

   a. Opening the application

   b. Defining Area to track

   c. Setting the click criteria

   d. Testing the mouse tracking using camera

   e. Testing click using defined criteria

5.2 System Test

5.2.1 Prerequisites to System Test

1. Test Plan
2. SRS
3. Test Scripts
4. Tester

5.2.2 Overview, Scope and Objective

System Testing will be carried out for Camera Mouse to ensure that required system functionalities conform to what has been defined in the FRS.

The System Test Scripts will cover the following:

A. System test scripts will provide coverage to all requirements documented in the Camera Mouse FRS.

B. System test script execution will be performed manually on the qualified test environment

   a. Tester instructions:

      i. Execute each step specified in the "Description" column in the system test script. Compare the outcome of each test step with the details listed in the "Expected results" column

      ii. After completion of System Test script execution, the tester is required to sign along with date in the "Script Executed By" section

   b. Test Script Deviation and Deviation Resolution

      i. For any test where the observed result does not match the expected result, will cause a deviation.

      ii. Any unresolved deviations will be documented with justification in the Validation Summary Report

   c. Suspension of Testing

      i. Testing will be suspended between the identification of a deviation and its resolution

   d. Reviewer instructions:

      i. Review the executed test scripts and ensure that the test script execution, documentation is correct and accurate.

C. The following modules of Camera Mouse application will be verified during the system testing:

   a. Opening the application

   b. Defining Area to track

   c. Setting the click criteria

   d. Testing the mouse tracking using camera

   e. Testing click using defined criteria

6 Document Management

If there is a need to update this document during the course of the project, a new version will be created and re-approved. All changes will be described in the document revision history section.

Software Engineering

Test Case 1

Test Case 1.0 (Camera Mouse Functionality test)

Test Case #: 1

System: **Facial motion tracking based mouse controller for disabled.**

Short Description: **Test the Camera mouse application**

Test Case Name: Check overall functionality of system

Design Date: **10/12/2021**

Page: **1 of 1**

Pre-conditions

The user has a windows or Linux based GUI OS.
The user's PC has a webcam support.

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Detect face of user	The system detects the face and place a square around it.	Pass	N/A
2	Head moves left	System detects the movement and move the cursor in the left direction.	Pass	N/A

3	Head moves right	System detects the movement and move the cursor in the right direction.	Pass	N/A
4	User closes eyes.	System detects the movement and the selected option of clicking is activated.	Pass	N/A
5	User presses the reinitialize button	Threshold is refreshed and sets a new threshold based after passing the start option again.	Pass	N/A

Test Report

S.no	Steps Executed	Steps Passed	Steps Failed
1	5	5	0